Customer Experiences Running Oracle & Open Source Databases with zVM

Sphinx Leadership Suite

David Simpson
IBM Washington Systems Center
Oracle Certified & Open-Source Database Specialist
simpson.dave@us.ibm.com

Ideas/Topics

- Large Memory Support
- Financial Lessons learned Implementing Oracle
- Backup and Recovery
- Ansible Automation/Golden Image
- I/O Demo – (audience help)
- AI Demo

z/VM Large Guest Memory Support Considerations (Early Adopters)

- z/VM supported limit: 2TB per LPAR, 4 TB with z/VM 7.2 + VM66173
- z/VM virtual machine size supported: 1 TB and **now 2 TB** (for Early Adopters with conditions)
- Maximum Oracle SGA (19.19 version) **1813.5 GB** (need memory for other functions) Oracle patch -> **34168505**

Live Guest Relocation

Relocating a large guest may take an extremely long time, not supported for Oracle

Reset Time

A guest reset (e.g., performed as part of LOGOFF or re-IPL processing) may take an extremely long time. Introduced minor changes that reduce the interval between guest LOGOFF initiation and subsequent re-LOGON

Helpful Commands and Tools

VMDUMP of large guest may run slowly to be viable and in any event is limited to dumping memory only up to 512GB.

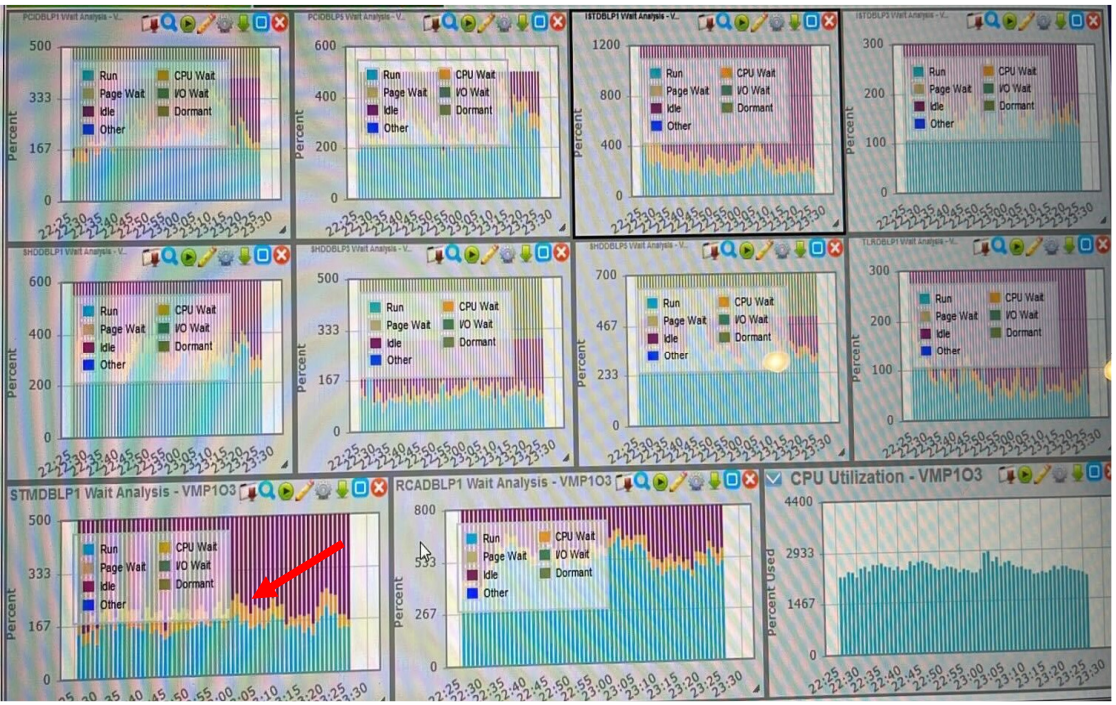
Small Guests

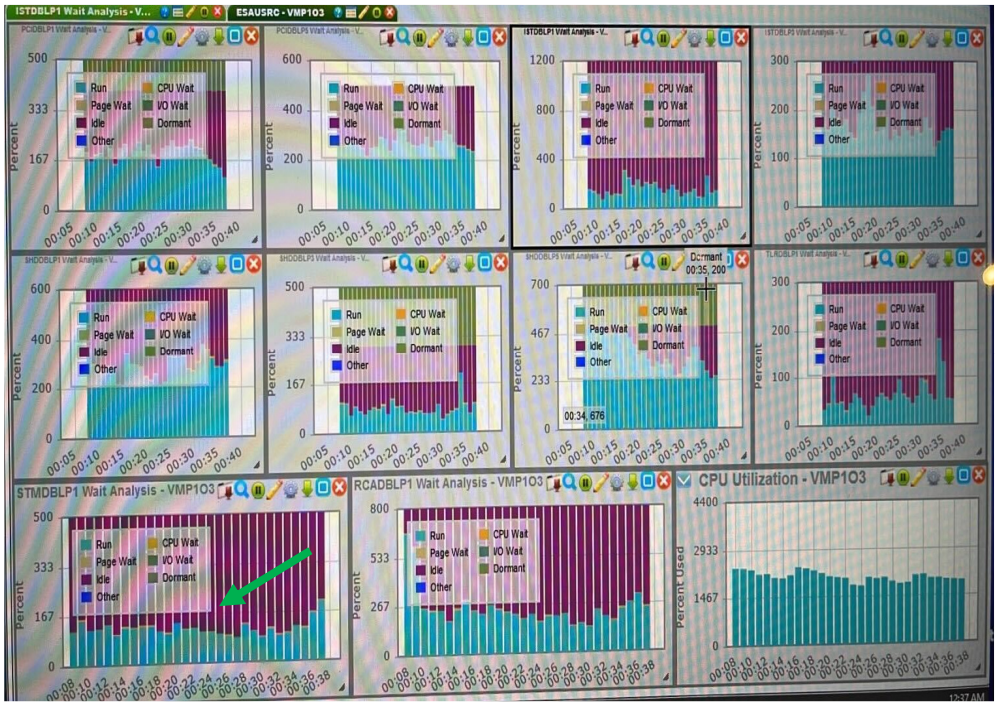
Smaller guest may be disadvantaged by larger one if memory is overcommitted.

Source: z/VM Limits https://www.vm.ibm.com/memman/gt1guest.html

Large Bank Oracle Experiences: Tuning CPU Requirements:

1. Make sure each guest has enough virtual CPUs to handle it's peak load (suggest cpuplugd to reduce / add cpu)
2. Check with z/VM support about changing from z/VM MODLEVEL 1 to MODLEVEL 0 is set (CP SET SYSCONTROL DISPATCH MODLEVEL 0 Reduces CPU wait in **yellow** below)
3. Fine-tune share settings of Linux guests





Client Case: Oracle RMAN Backup Traffic

Separation of backup traffic from user network (if possible separate physical network)

- Symptom: Deep dispatch queues and long dispatch delays for guest virtual CPUs. Slow down in vswitch as transfers are done by CP under the dispatch of the guest virtual CPU who initiated them. Dedicated OSA does not have these properties due to QDIO assist .
- Second vNIC, even connected to same VSWITCH, provides ability to use different MTU sizes for different traffic types & configure priority of the different vNICs

OPTIONS="layer2=1 portno=0 **buffer_count=128**“ - Default is buffer_count=64

ping with packet size (-s) of 8972, preventing fragmentation (-M do) to confirm MTU 9000 (28 bytes overhead)

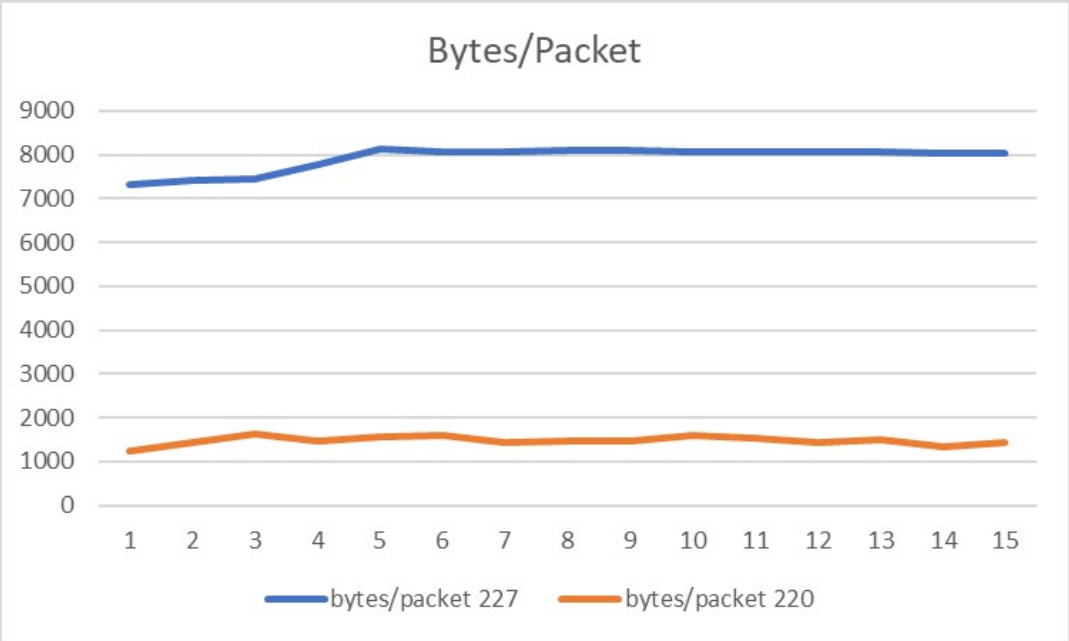
End Result:

TNS pings (no deviation for users) and backup traffic throughput (less cpu usage, less DIAG 9C...)

we saw 904mB/s (7.2 gb/s) on a 10Gb OSA, with no constraints imposed by the bit rate. (OSA kept up just fine, no uplink TX discards.)

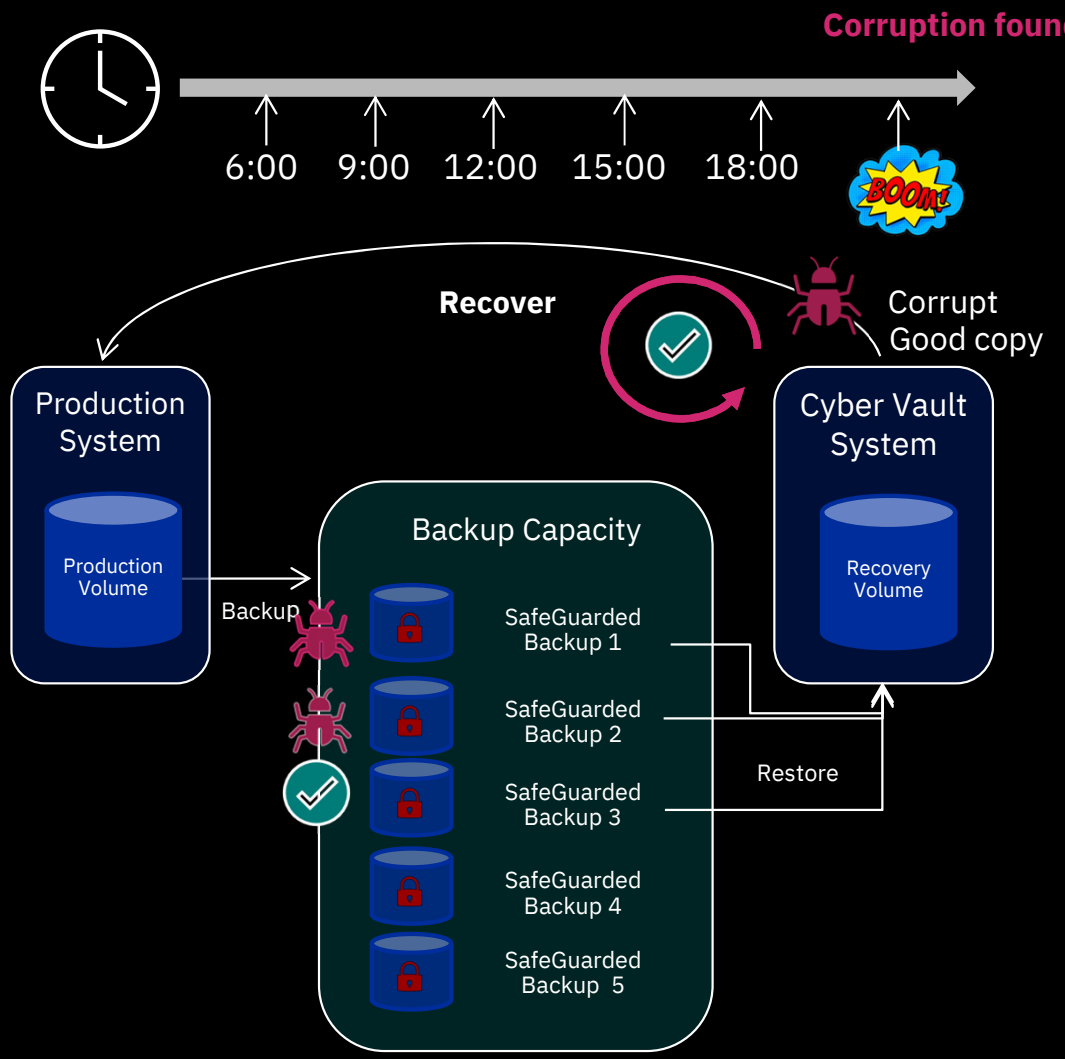
Separating Backup Traffic from User Network

- Backup data rate went from 419 mB/sec to 904 mB/sec. Increased data rate by 116%.
- Bytes/pkt went from 1518 to 8218 - guest and host are more efficient
- CP CPU time per mB transmitted went from 784 us to 380 us.
- Total CPU time per mB transmitted went from 10250 us to 4368 us. 57% reduction in CPU use per mB sent. Host (CP) is operating much more efficiently.



IBM Spectrum Virtualize provides Safeguarded Copy

- **Logical Corruption Protection** to prevent sensitive point in time copies of data from being modified or deleted due to errors, destruction or ransomware
- Up to **15864 objects** to provide **immutable Safeguarded copies** of production data stored in Safeguarded backup capacity known as a Child Pool
  - Not directly accessible to any server or application
- Data is accessible *only* after a Safeguarded copy is **recovered to a separate recovery volume**.
- **Proactive monitoring** for signs of attack
  - Identify Safeguarded Volume to recover based on time index of identified attack
- Recovery volumes are used for:
  - Data validation
  - Forensic analysis
  - Restoration of production data



Spectrum Scale 5.0 with Oracle Backups Test Case:

Test Case:

Seven snapshots representing everyday. Sunday, a level-0 (FULL) backup is taken. Monday to Saturday incremental backup and merge is done.

Advantages backup scheme:

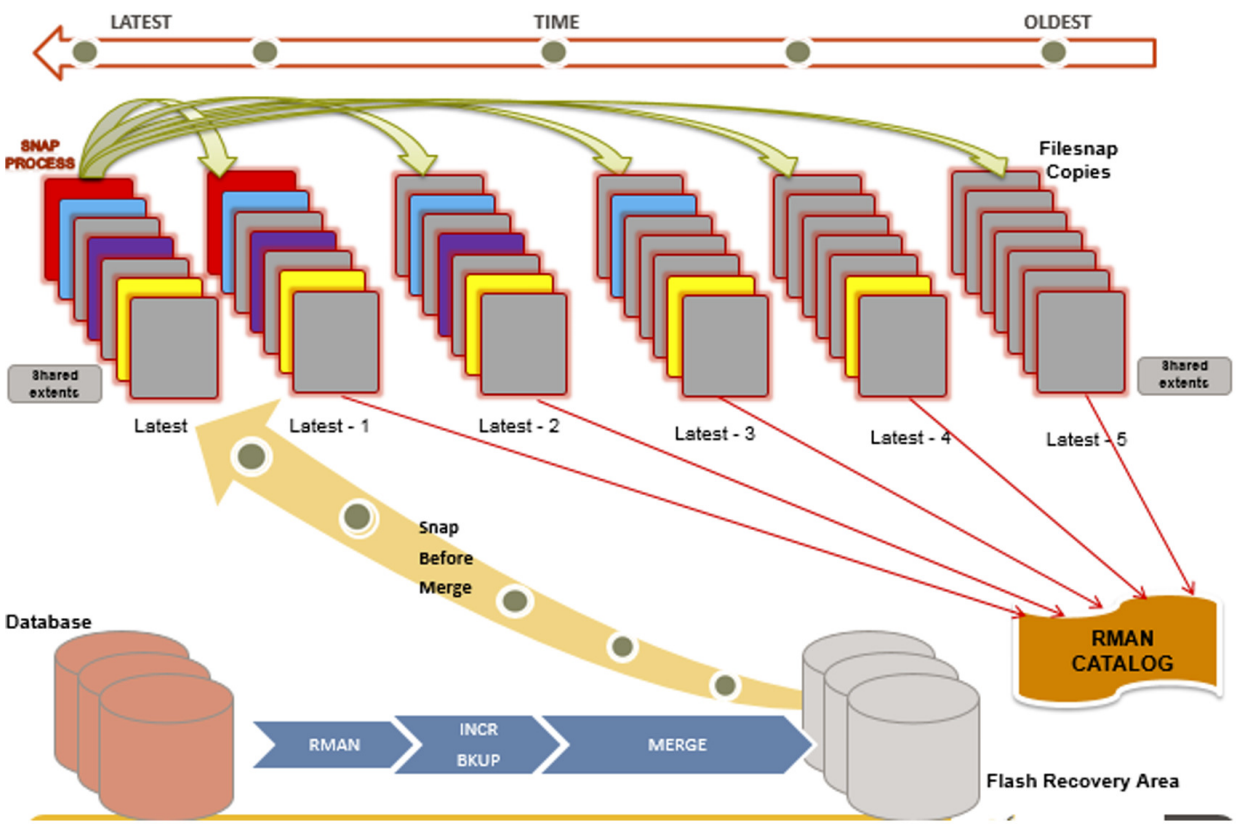
- Daily backup window is reduced. Only changed blocks from previous day are backed up.
- After snapshot, incremental backups are merged back with the level-0 backup. After the merge, we have full backup.
- Changed blocks since previous incremental backup, are pushed into snapshot (**Block Push to snap**). This means we have Sundays and Mondays level-0 backup. For a week, we have seven FULL backups.
- For restore, since we always have a current level-0 backup copy, no need to merge, apply lots of archivelogs during restore (faster RTO)
- Recovery time fast and customers can meet recovery time objective (RTO) goals.
- The workload we ran against the database inserts one million rows; updates one million rows, update indexes.
- The table below shows the amount of disk storage saved, **The original DB Size is 118 GB**

SnapShots	Backup Size	Incr Backup Size	Block Push to snap
0	118 GB	None	None
1 (snp_0717@16H56M)	125 GB	50 GB	6.8 GB
2 (snp_071717H19M @)	131 GB	63 GB	6.1 GB
3 (snp_0717@17H35M)	137 GB	54 GB	6.1 GB
4 (snp_0717@18H01M)	144 GB	60 GB	6.2 GB
5 (snp_0717@18H19M)	150 GB	63 GB	6.2 GB
6 (snp_0717@19H01M)	157 GB	62 GB	6.3 GB
7 (snp_0717@19H29M)	164 GB	60 GB	6.1 GB

Oracle RMAN Merge Incremental

Supported Backup, Restore and Recovery Operations using Third Party Snapshot Technologies (Doc ID 604683.1)

Integrating IBM Spectrum Scale snapshots with Oracle Recovery Manager incremental backups



Source: https://www.ibm.com/support/pages/system/files/inline-files/ORA_DB_IBM_Z_Spectrum_Scale_Malige_23MAY18_final.pdf

Active File Management – Motivation (example)

An organization's main data center has a large (Spectrum Scale) file system, `/scale/fs1`, and a main compute / application cluster.

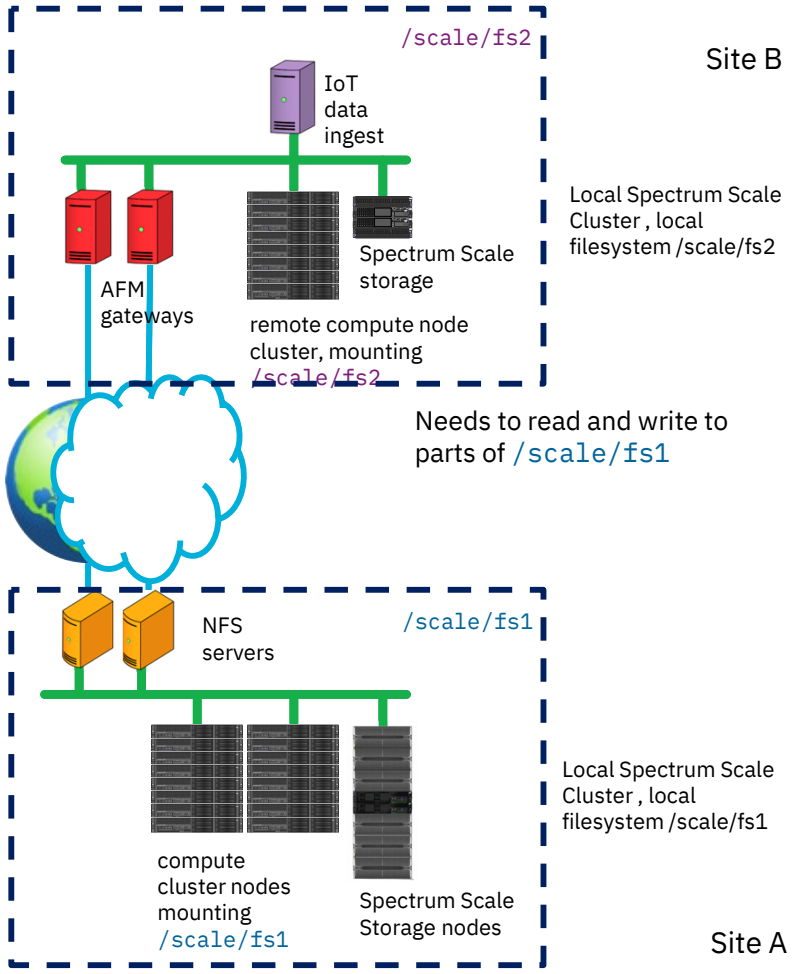
A secondary site is doing its own computation, with data stored locally in `/scale/fs2`.

(It might be remote because of special data acquisition or ingestion needs.)

But the remote site also needs data from `/scale/fs1` and needs the main compute cluster to work on data it has ingested (ideally stored in `/scale/fs1`).

We can't afford the latency of accessing data stored across the world, and we need the performance of Spectrum Scale.

Active File Management (AFM) lets us set up filesets in the remote cluster as caches of parts of `/scale/fs1`, using NFS transport.



Overview of AFM caching – file storage

AFM **cache filesets** cache data from a target or **home**, using several modes:

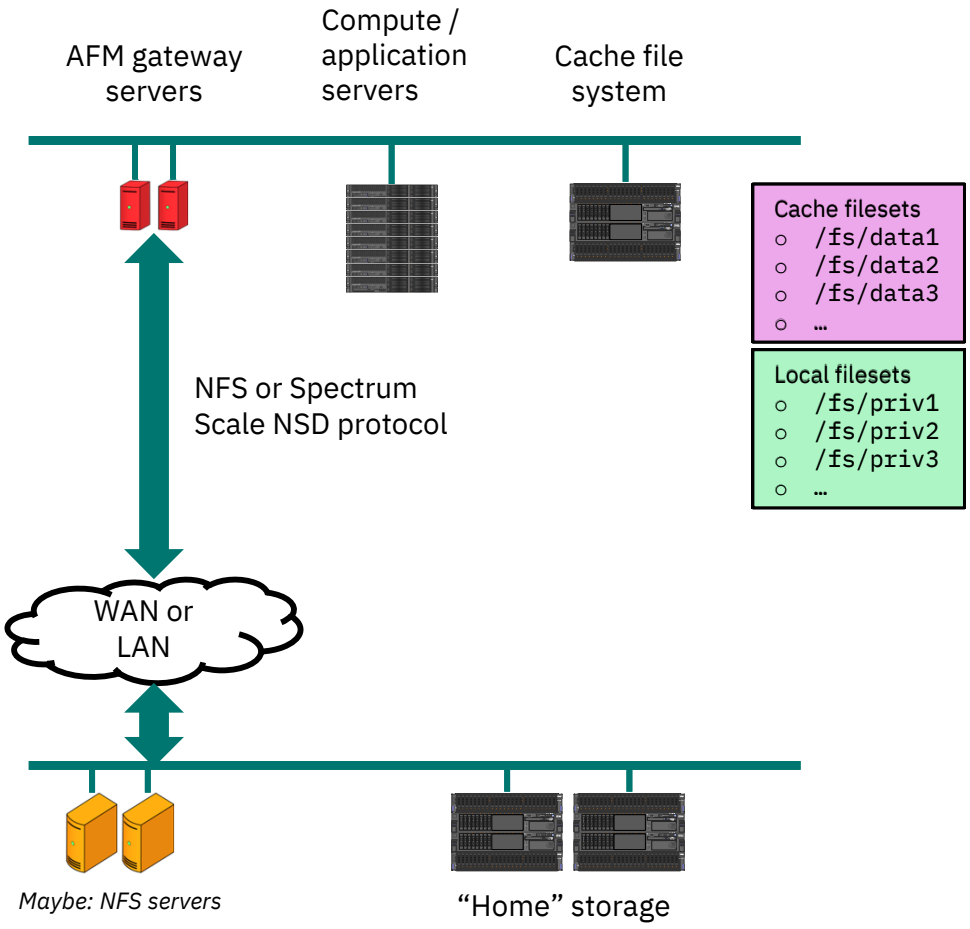
- **Read-only** – A refreshable read-only cache of the home
- **Local update** – Read-only cache, but local-only changes allowed
- **Single writer** – Only the single cache; it alone may update home. AFM **primary** and **secondary** filesets are based on single writer.
- **Independent writers** – multiple caches may update same home

Gateway servers maintain the freshness of caches and send updates to the home. They queue updates even if the link to home is down.

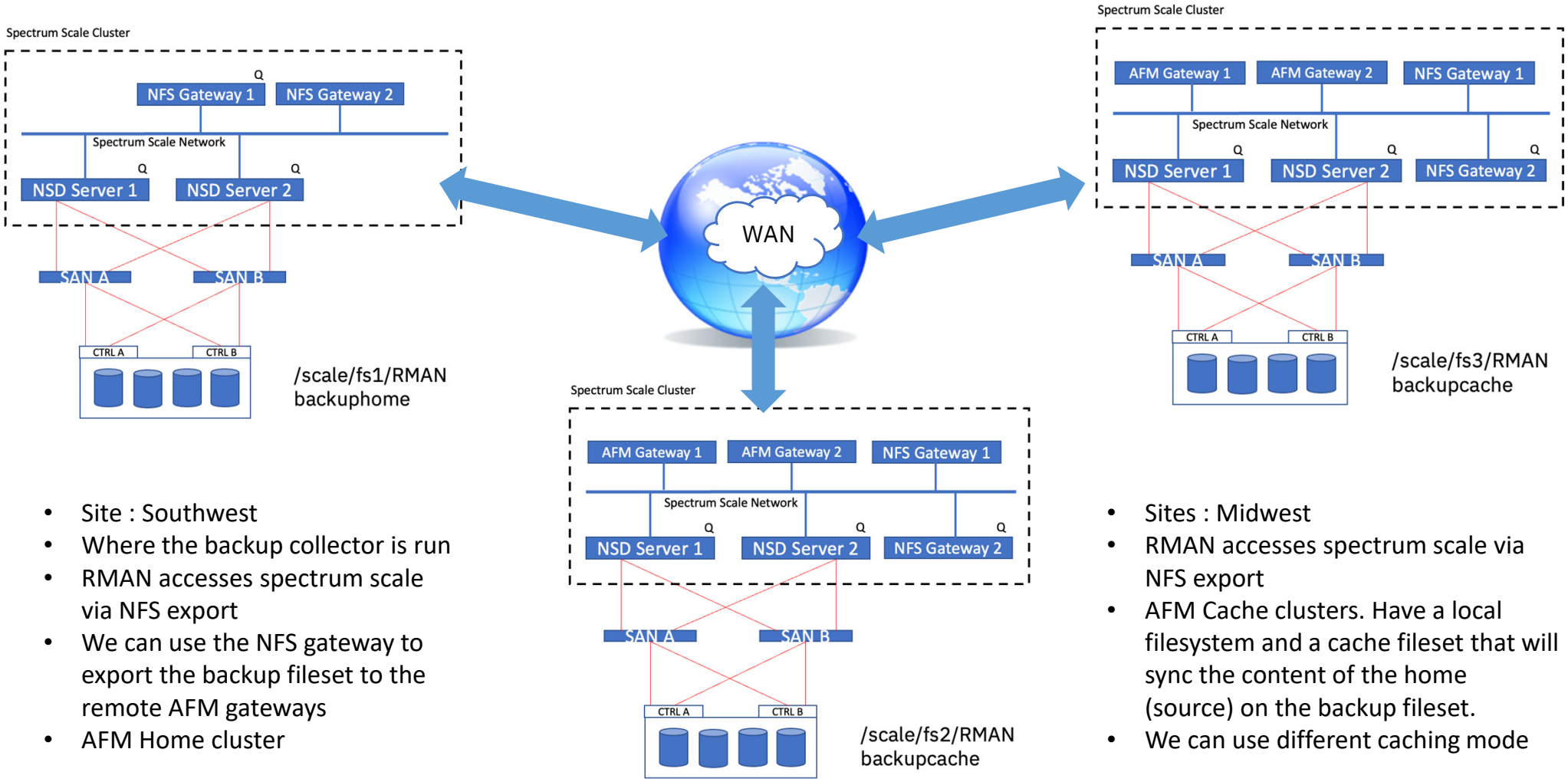
A **home** is a directory tree being cached by AFM somewhere else. The home is unaware of the cache and need not even be in Spectrum Scale.

Transport is **either NFS or Spectrum Scale's NSD protocol**.

Cache revalidation is on demand, and the granularity of updates is the file system block.



Proposed High Availability Spectrum Scale Architecture

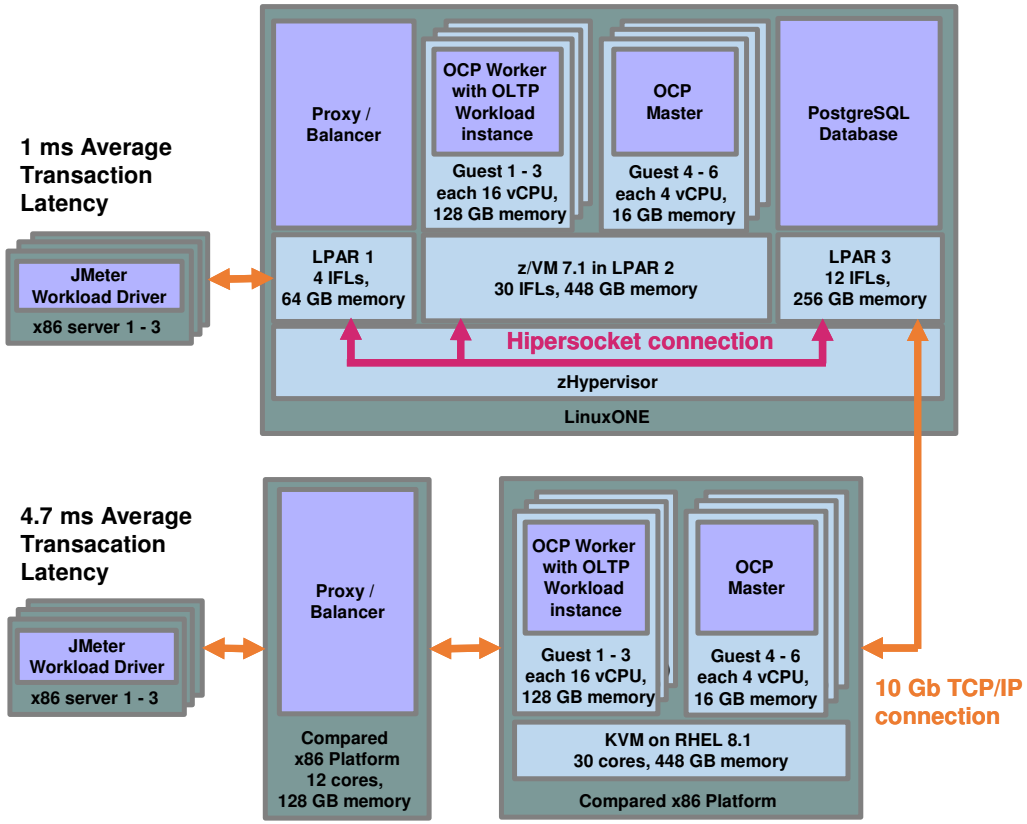


- Site : Southwest
- Where the backup collector is run
- RMAN accesses spectrum scale via NFS export
- We can use the NFS gateway to export the backup fileset to the remote AFM gateways
- AFM Home cluster

- Sites : Midwest
- RMAN accesses spectrum scale via NFS export
- AFM Cache clusters. Have a local filesystem and a cache fileset that will sync the content of the home (source) on the backup fileset.
- We can use different caching mode

OLTP workload on OpenShift (OCP) with database co-location on LinuxONE versus remote database access from x86 Skylake

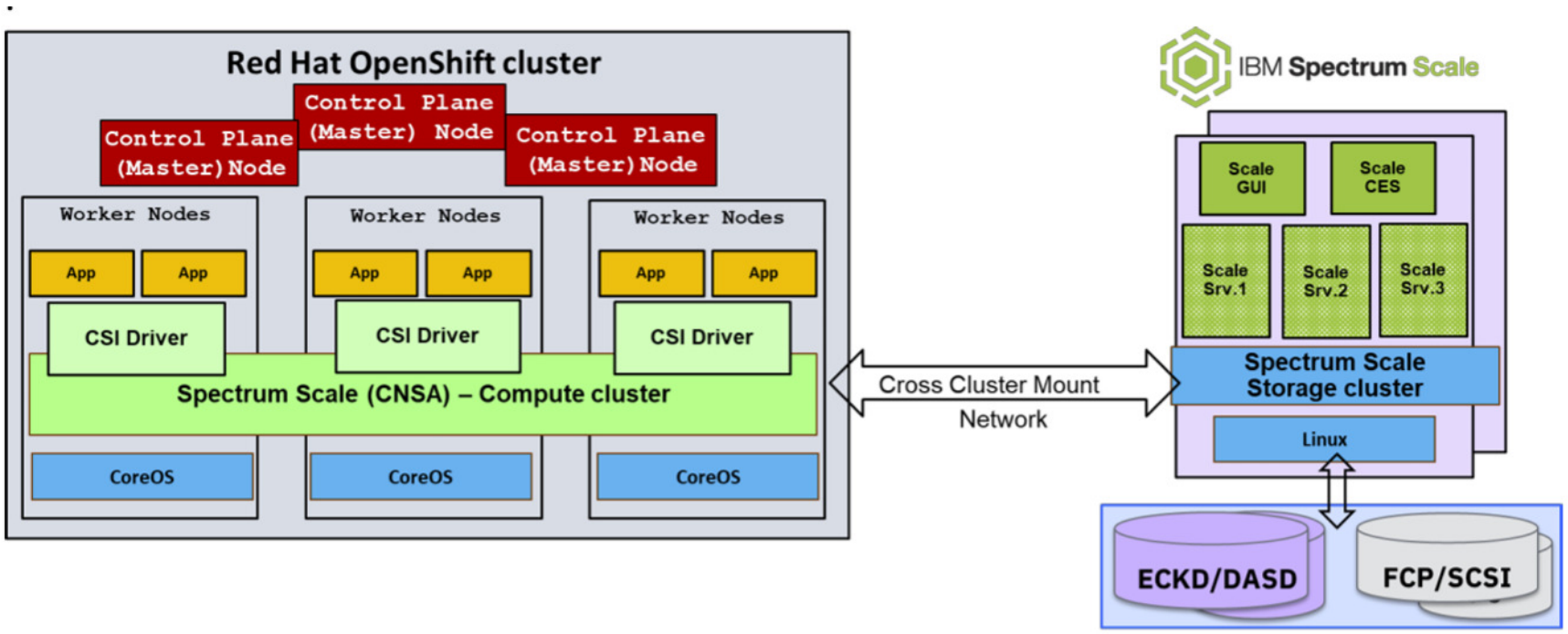
Run an OLTP workload on OpenShift Container Platform 4.4 with up to 4.7x lower latency co-located to the used database on LinuxONE using a Hipersocket connection versus on compared x86 platform using a 10 Gb TCP/IP connection to the same database



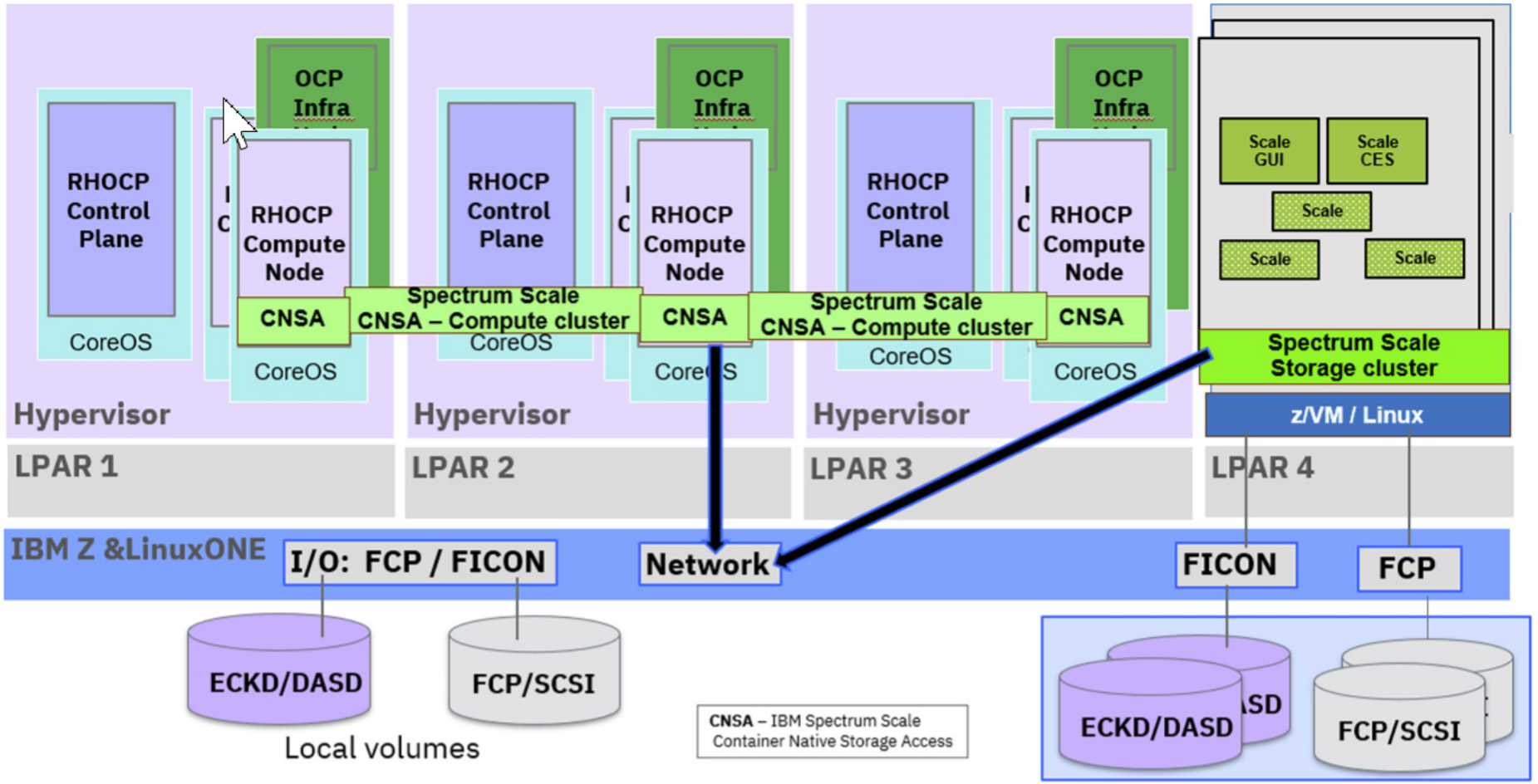
DISCLAIMER: This is an IBM internal study designed to replicate banking OLTP workload usage in the marketplace deployed on OpenShift Container Platform (OCP) 4.4.12 on LinuxONE T01 using z/VM versus on compared x86 platform using KVM accessing the same PostgreSQL 12 database running in a LinuxONE T01 LPAR. 3 OLTP workload instances were run in parallel driven remotely from JMeter 5.2.1 with 16 parallel threads. Results may vary. LinuxONE T01 configuration: The PostgreSQL database ran in a LPAR with 12 dedicated IFLs, 256 GB memory, 1TB FlashSystem 900 storage, RHEL 7.7 (SMT mode). The OCP Master and Worker nodes ran on z/VM 7.1 in a LPAR with 30 dedicated IFLs, 448 GB memory, DASD storage, and Hipersocket connection to the PostgreSQL LPAR. x86 configuration: The OCP Master and Worker nodes ran on KVM on RHEL 8.2 on 30 Skylake Intel® Xeon® Gold CPU @ 2.30GHz with Hyperthreading turned on, 448 GB memory, RAID5 local SSD storage, and 10Gbit Ethernet connection to the PostgreSQL LPAR.

IBM Spectrum Scale on IBM Z components:

The connection to RHOCP is implemented as cross-cluster mount, and in a clustered mode with the Container Native Storage Access (CNSA) components.



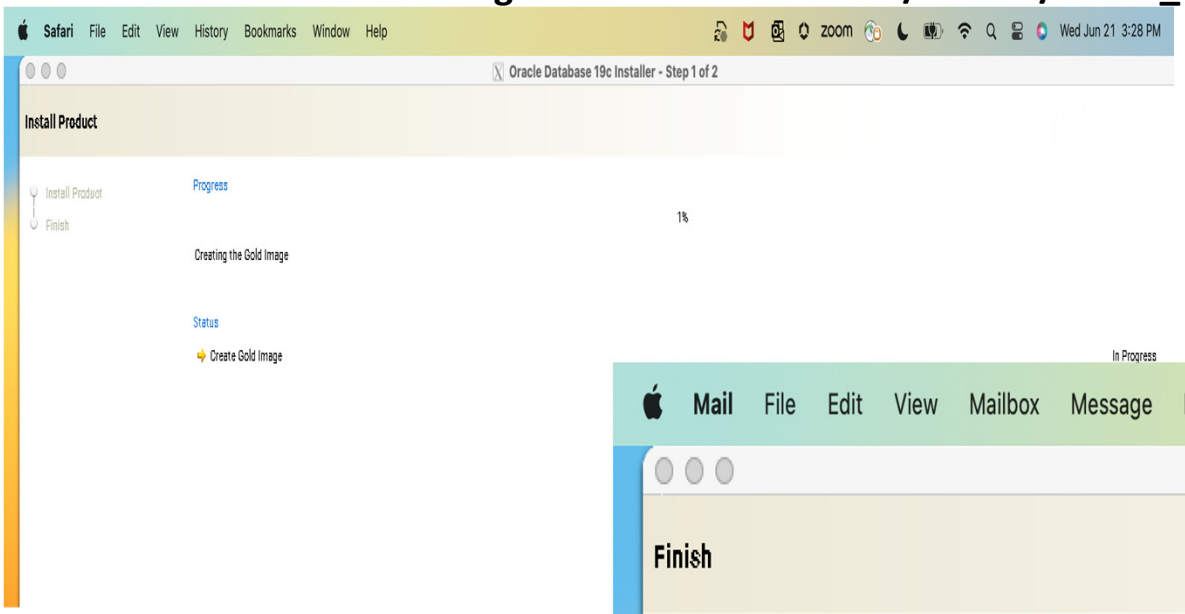
IBM Spectrum Scale on IBM Z components:

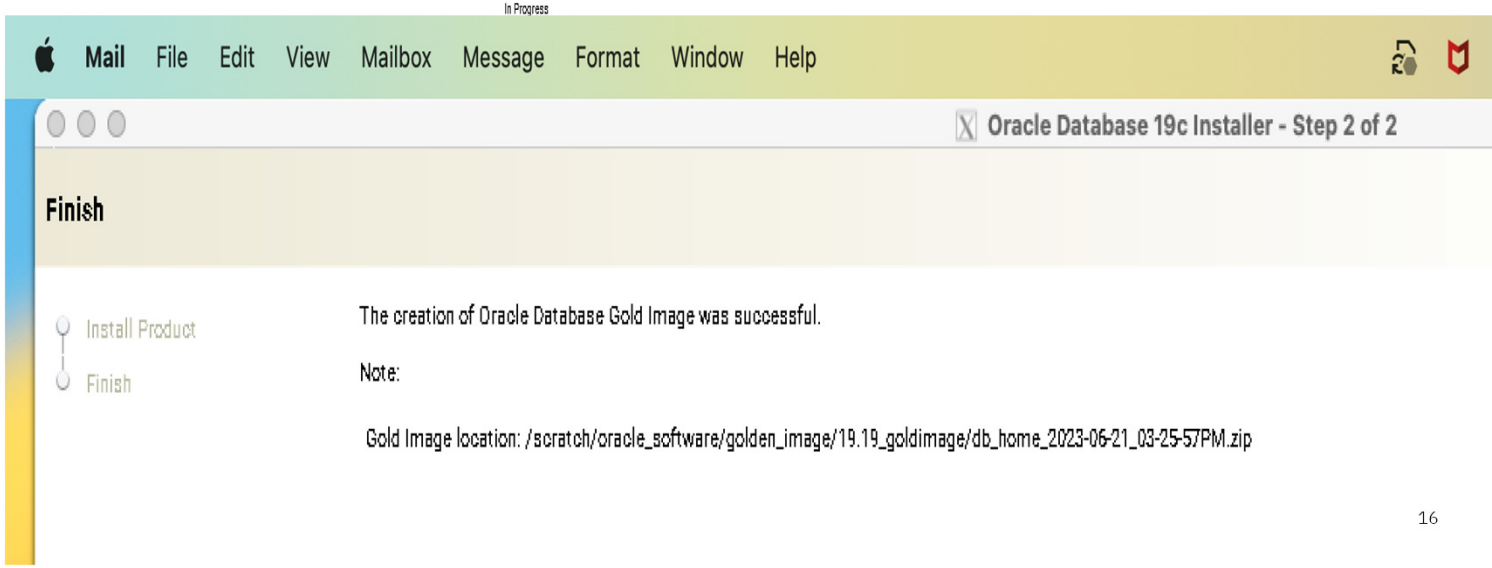


Oracle Golden Image – Oracle 19.19+

Do the installation once and then copy the homes!

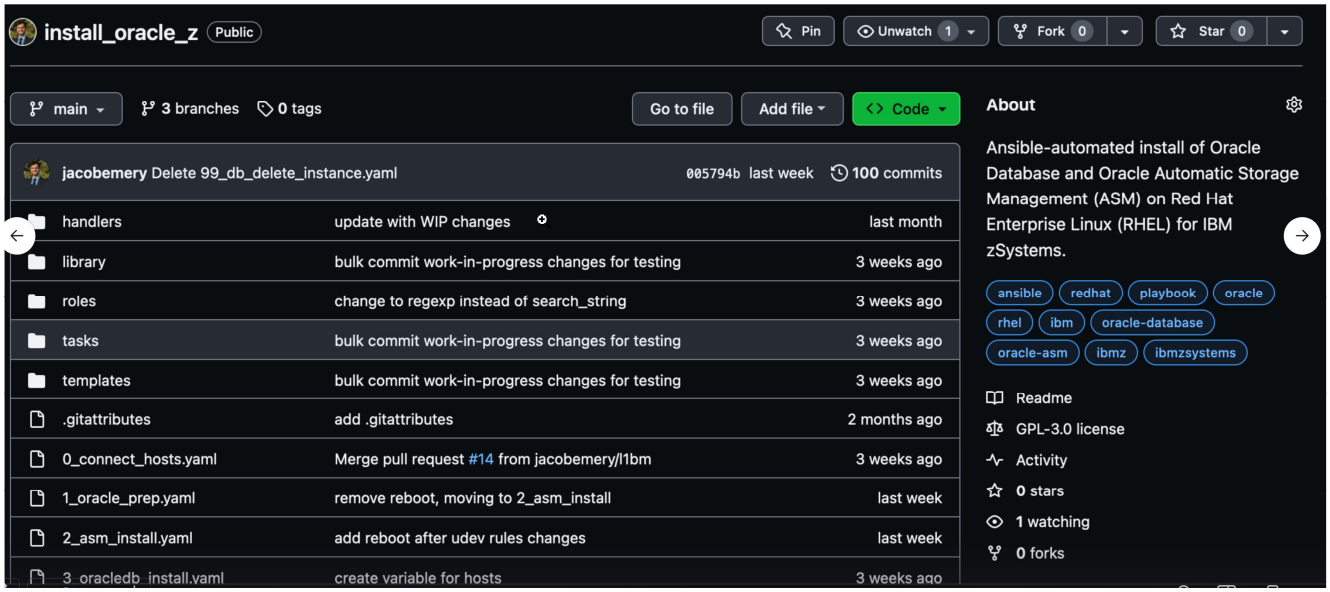
runInstaller -createGoldImage -destinationLocation /scratch/oracle_software/golden_image/19.19_goldimage





Ansible Automation

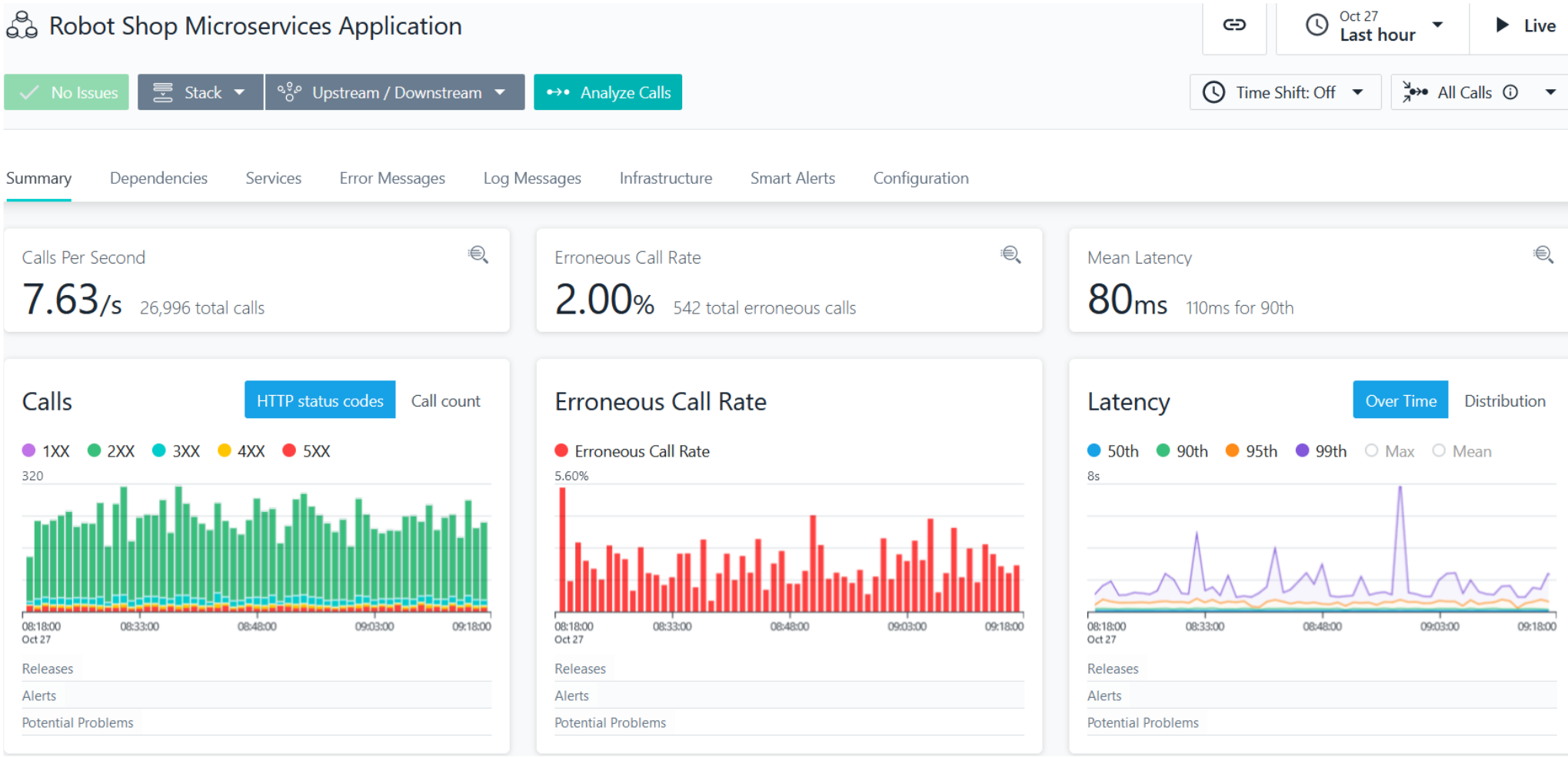
- Oracle Installation’s (using Golden Image files)
- Oracle Patching or Installing a new Golden Image
- Spectrum Scale backup configuratons
- Oracle RMAN backup configurations
- Automated server build (one node RAC) for Spectrum Scale immutable backup restore activity
- Ansible Tower to run scripts
- Grandma’s white bean chili



https://github.com/jacobemery/install_oracle_z

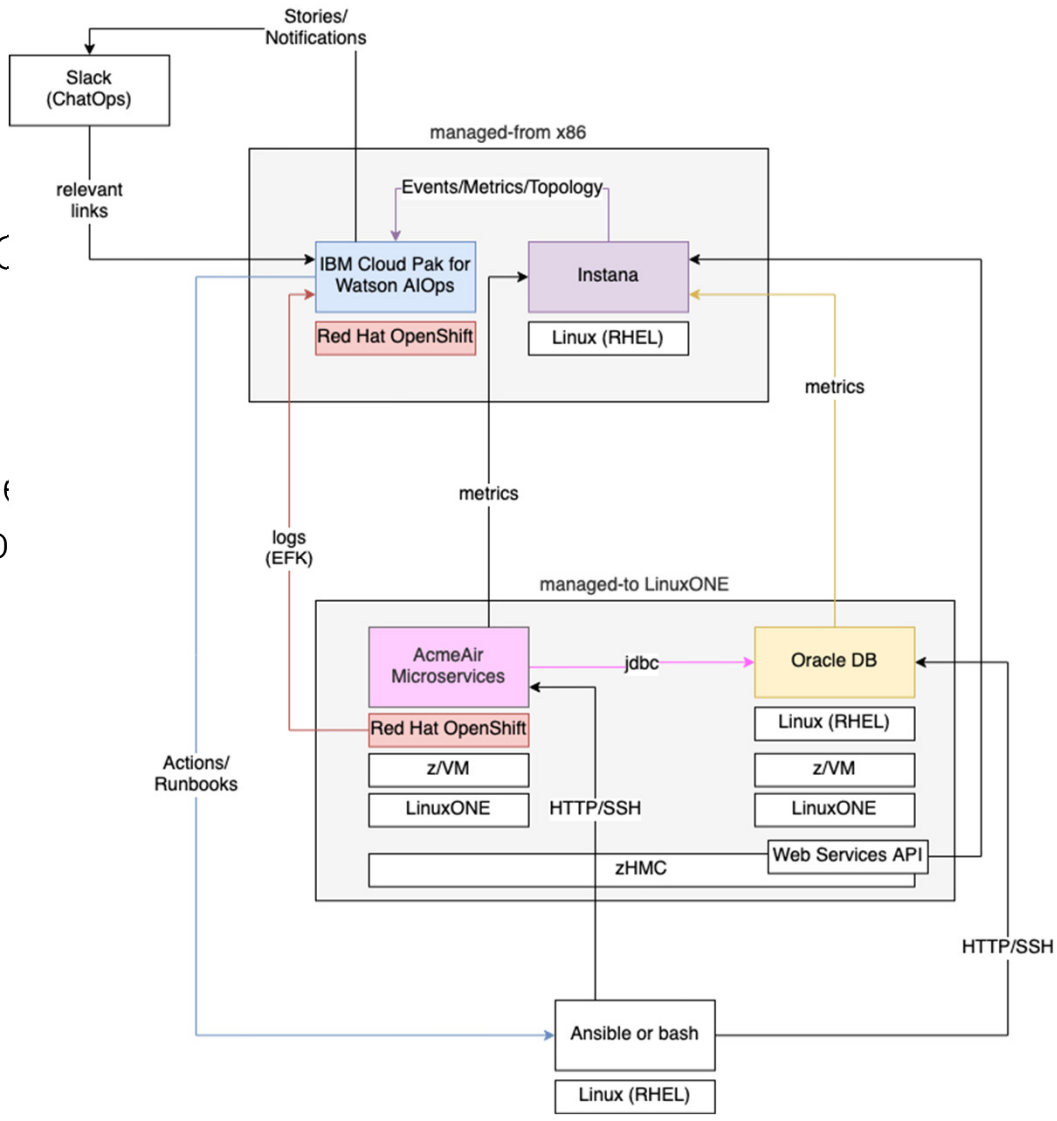
```yaml
- name: Make Grandma's white bean chili
  hosts: home_kitchen
  vars_files: ingredients.yaml
  tasks:

    - name: Gather ingredients
      community.traderjoes.shop:
        name: "{{ item }}"
        state: fresh
      loop: bean_chili

    - name: Wash veggies
      ansible.appliances.sink:
        temp: cold
        state: cleaned
      loop: bean_chili

    - name: Chop vegetables
      ansible.tools.knife:
        name: "{{ item.veggie }}"
        state: "{{ item.method }}"
      with_items:
        - { veggie: white_onion, method: diced }
        - { veggie: jalapeños, method: sliced }
        - { veggie: gold_potatoes, method: chopped
}
        - { veggie: garlic, method: minced }
```

Monitor Your Applications – IBM Instana



Demo integration of Watson AI, Ansible, Oracle and Mongo

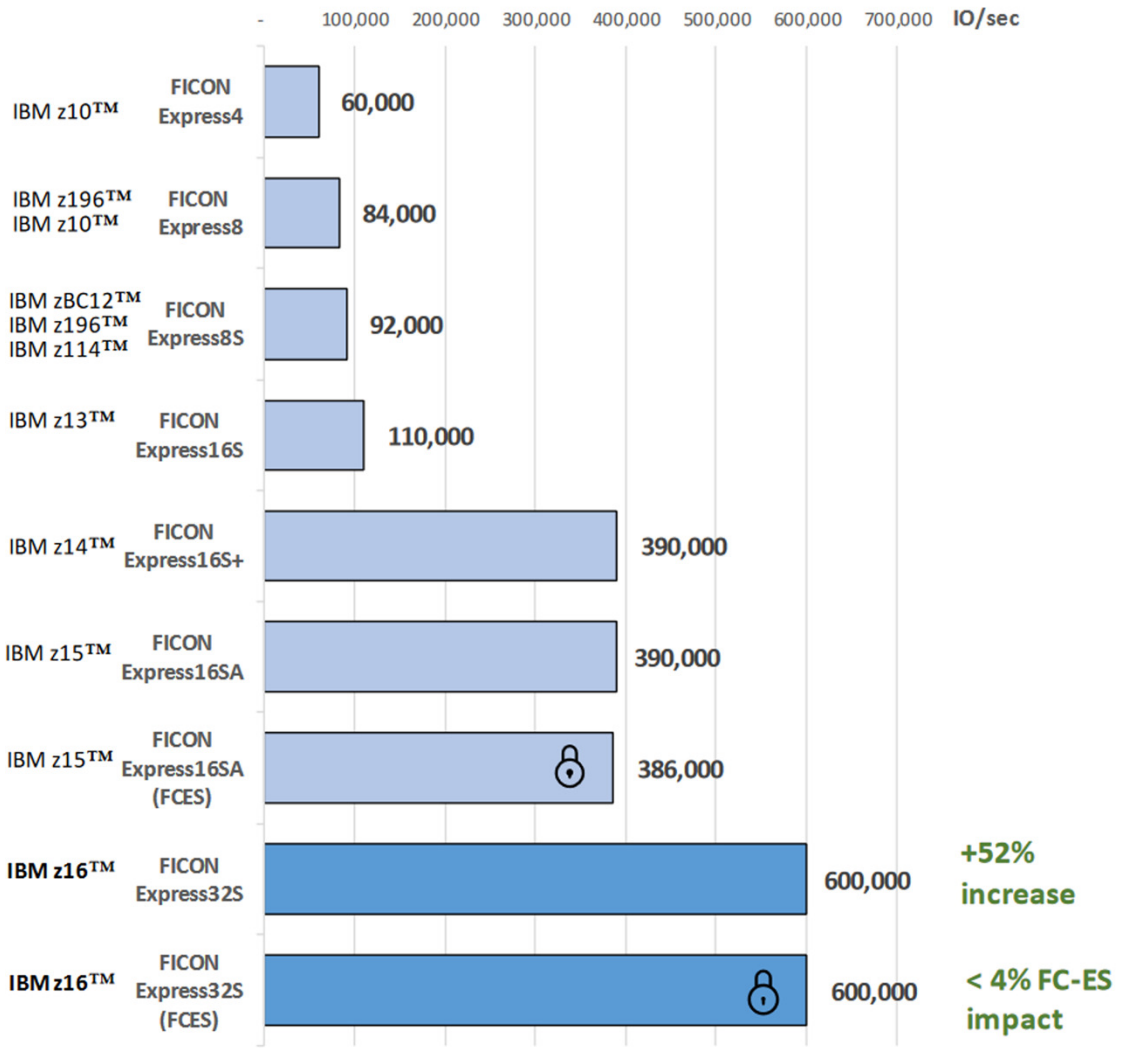
- Instana and IBM Cloud Pak for Watson AIOps managing-to OpenShift and Oracle DB on LinuxC
- Ansible playbooks to manually or automatically remediate OCP/Oracle DB issues
- Slack ChatOps to quickly notify SREs and provide links to the active alerts and remediation playbo
- Available for viewing on our TechBytes series online https://www.crowdcast.io/e/techbytes/6



High Performance I/O

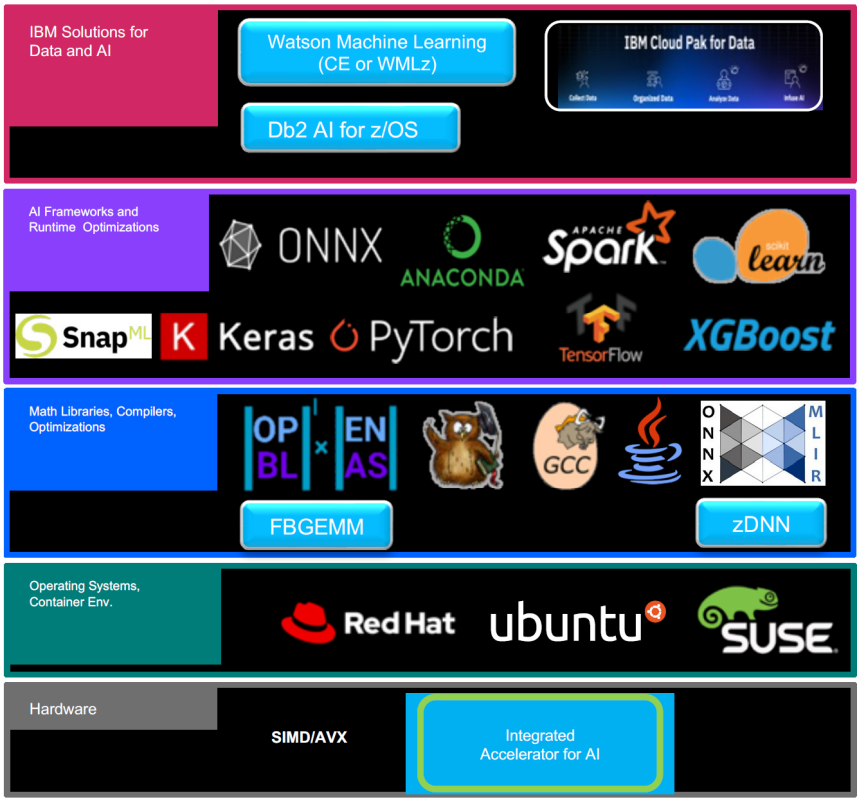
I/O Demo:

Customer Experience:



Source: https://www.ibm.com/support/pages/system/files/inline-files/IBM%20z16%20FEx32S%20Performance%202022.pdf

Designed for business insights and operational excellence



IBM Z strategy takes a holistic approach to AI:

- Enable popular open-source data science packages on platform
- Optimize libraries and compilers to leverage IBM Z architecture investments
- Leverage open source directly – or couple with the best of IBM's AI offerings

- https://github.com/IBM/zDNN
- https://github.com/onnx/onnx-mlir (C/C++ or Java)
- https://github.com/IBM/onnx-mlir-serving (Nvidia Triton backend)
- https://ibm.github.io/ibm-z-oss-hub/main/main.html

CCF Inference Demo Architecture

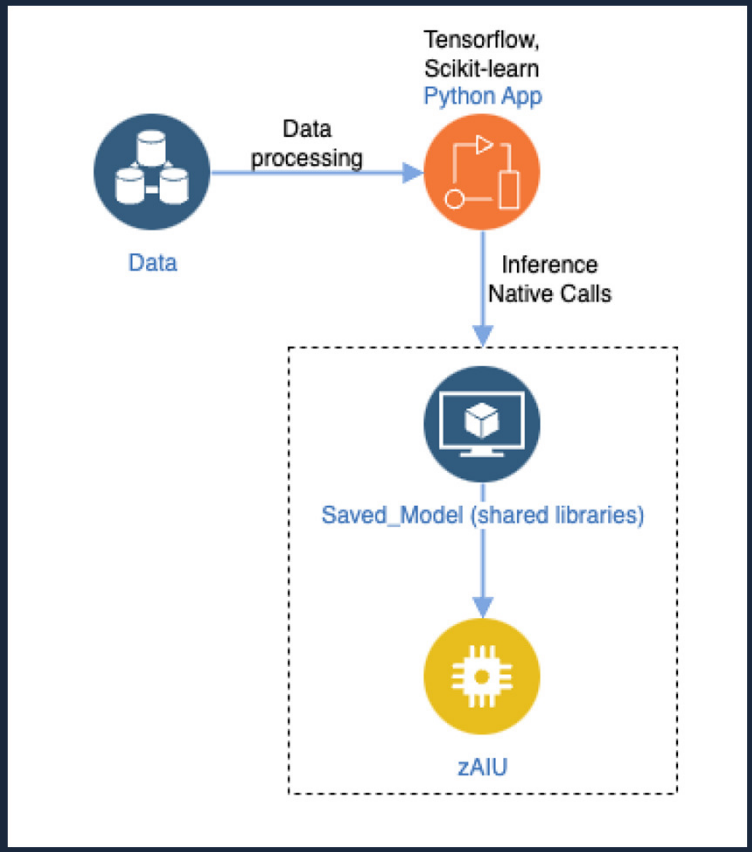
Fraud Detection Inference

- Pre-trained Fraud Detection models by TensorFlow that are saved in saved_model format.
- Runs inference against 100K credit card transactions.
- The models use LSTM/GRU and Dense layers.

Environment:

- LinuxONE IV
- Tensorflow v2.9.1
- Python

- Demo: https://ibm.ent.box.com/s/dgt13upmae9kyqf7batvyucyi007xggc

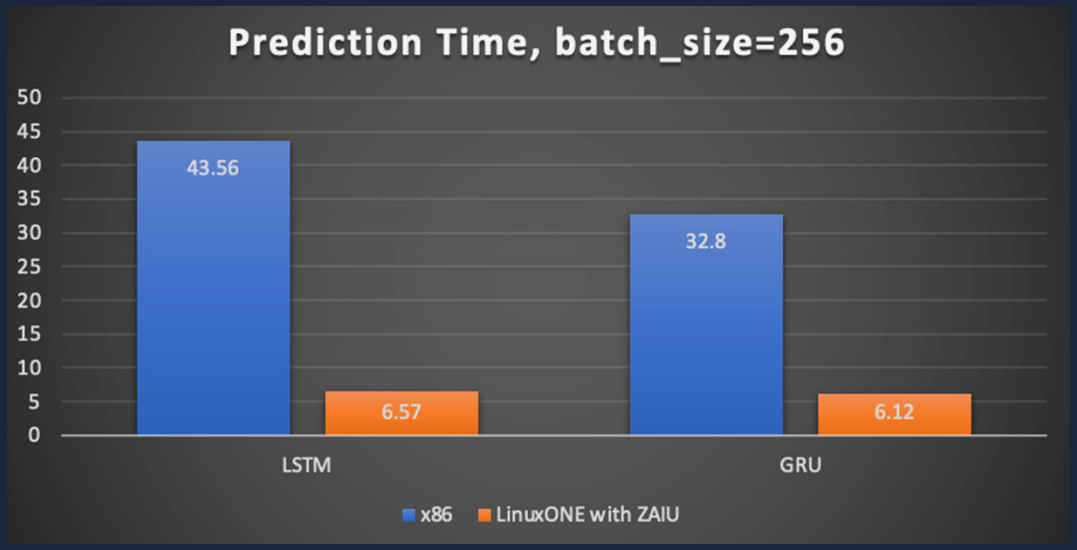


Credit Card Fraud Detection Performance

LinuxONE Prediction Time with batch_size=256 is **6.5ms.**

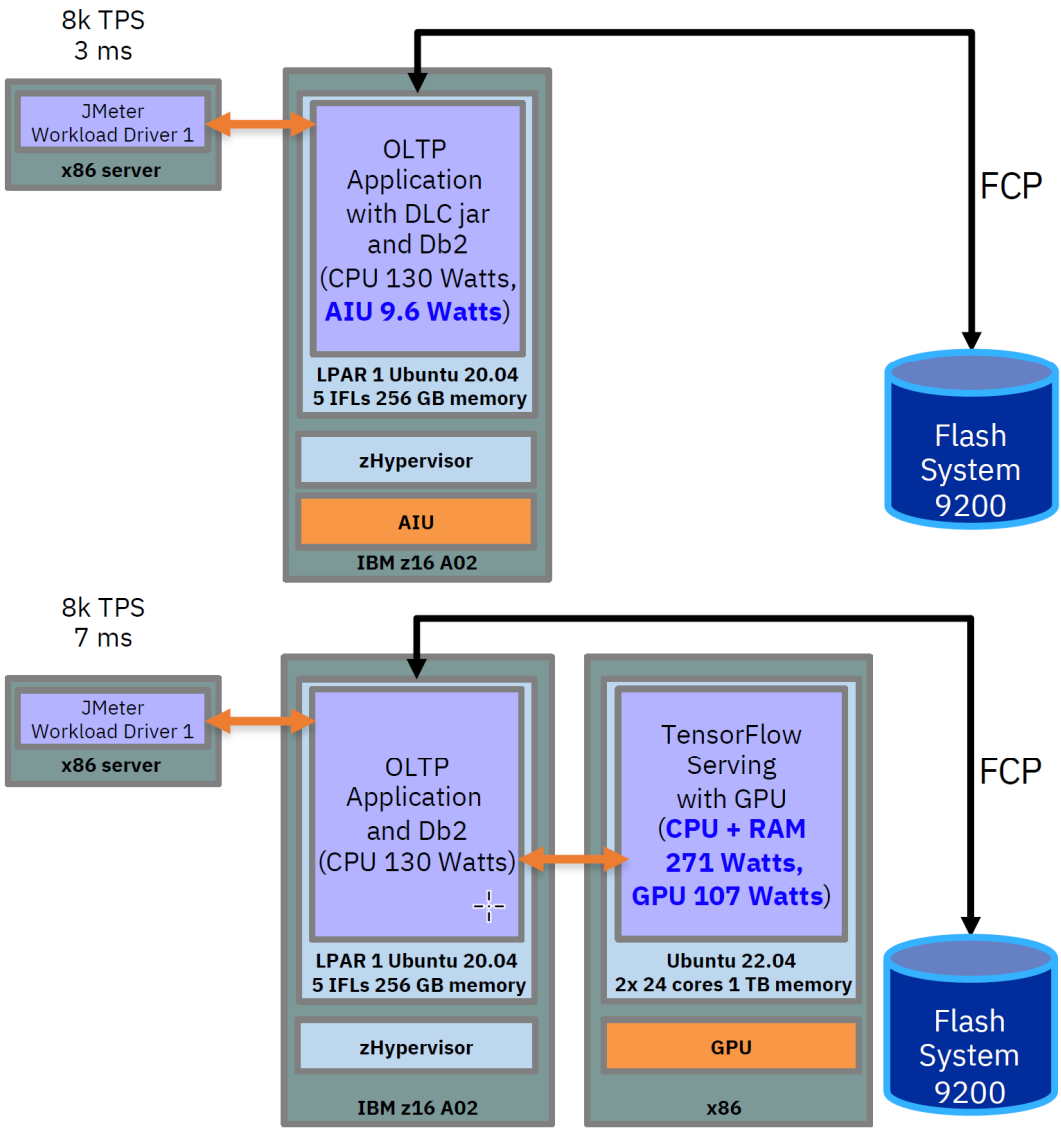
6.6x less prediction time compared to x86

```
Batch count : 390
Generate component times:
Process.TRANSFORM = 17.277
Process.MAKEDATA = 2.837
Process.MAKETARGET = 0.0
Process.TRANSPOSES = 0.811
Average inference time for each instance:
2.5657958422715847e-05
Average inference time for each batch with
batch_size=256: 0.006568437356215257
Confusion Matrix :
 [[99723 1]
 [ 1 115]]
F1 score : 0.99
Time taken for generating : 20.93
Time taken for prediction : 2.56
Total time taken for inference : 23.66
```



Integrated Accelerator for AI on IBM LinuxONE vs x86 with GPU Inferencing (Telum)

On IBM LinuxONE IV, reduced the energy consumption by 39x using the Integrated Accelerator for AI to process inference operations of an OLTP workload versus running inference operations remotely on a compared x86 server using an NVIDIA GPU



DISCLAIMER: Energy reduction includes only the inference operation processing but not the entire OLTP processing. Results based on IBM internal tests running an OLTP workload with credit card transaction using the Credit Card Fraud Detection (https://github.com/IBM/ai-on-zfraud-detection) model on IBM Model 3932 using the Integrated Accelerator for AI to process inference operations vs running the OLTP workload (https://github.com/IBM/megacardstandalone) on IBM Model 3932 with remote inferencing on a x86 server running Tensorflow serving. IBM Model 3932 configuration: Ubuntu 20.04 in an LPAR with 6 dedicated IFLs, 256 GB memory, and IBM FlashSystem 9200 storage. x86 configuration: Ubuntu 22.04 on 2x 24 IceLake Intel® Xeon® Gold CPU @ 2.80GHz with Hyperthreading turned on, 1 TB memory, local SSDs, NVIDIA® A40 GPU, UEFI maximum performance profile enabled, CPU P-State Control and CStates disabled. Results may vary.